

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО”**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ  
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ М. В. Грайворонський  
\_\_\_\_\_ (підпис) \_\_\_\_\_ (ініціали, прізвище)  
“ ” 2019 р.

**Дипломна робота**

освітньо-кваліфікаційного рівня “бакалавр”

з напрямку підготовки 6.040301 «Прикладна математика»

на тему «Знаходження оцінок статистичних параметрів фону»

Виконав студент 4 курсу групи ФІ-51

Макаренко Сергій Олегович

Керівник к. т. н. Андрій Миколайович Родіонов

Рецензент

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_

## РЕФЕРАТ

Дипломна робота містить 42 сторінки, 22 ілюстрації, і 19 джерел літератури.

Однією з проблем комп'ютерного зору є відновлення фону. Задача відновлення фонового зображення лежить в основі алгоритмів пошуку об'єктів та слідкуванню за ними.

Об'єктом дослідження є статичне відеозображення з рухомими об'єктами.

Предметом дослідження є оцінки параметрів ймовірнісних моделей.

Метою даної роботи є аналіз алгоритму знаходження оцінок максимальної схожості параметрів ймовірнісних моделей та отримання за його допомогою статистичних параметрів фону.

Для досягнення мети було використано

- MoG для моделювання розподілу.
- Рівномірний розподіл для кластеризації значень об'єктів.
- ЕМ для знаходження оцінок параметрів розподілів.

Ітеративні алгоритми, ем алгоритм, гауссовий розподіл, багатовимірні множини, MoG, рівномірний розподіл

## **ABSTRACT**

Graduate work contains 42 pages. 22 illustrations, and 19 sources of literature.

One of the problems of computer vision is the background recovery. The problem of restoring the background image is the basis of the search algorithms of objects and the tracking of them.

The object of the research is a stationary video image with moving objects

The subject of the study is to evaluate the parameters of probabilistic models.

This study aims is to analyze the algorithm of finding the estimates of the maximum similarity of the parameters of probabilistic models and obtaining using it the statistical background parameters.

To perform the study was used

- MoG for distribution modeling.
- Uniform distribution for clustering of object values.
- EM to find estimates for the distribution parameters.

Iterative algorithms, em algorithm, Gaussian distribution, multidimensional sets, MoG, uniform distribution

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ .....	8
1 Попередні роботи присвячені відновленню параметрів фону та слідкуванню за об'єктами .....	9
1.1 Базове моделювання фону .....	10
1.2 Кластеризування .....	11
1.3 Статистичні методи .....	16
Висновки до розділу 1 .....	21
2 Аналіз роботи відомих алгоритмів .....	22
2.1 Базове моделювання фону .....	22
2.2 Моделювання фону базоване на кластеризуванні .....	25
2.3 Статистичні методи відновлення фону .....	27
Висновки до розділу 2 .....	31
3 Розробка алгоритму відновлення статистичних параметрів фону .....	32
3.1 Підсумки з проаналізованих відомих алгоритмів .....	32
3.2 Запропонована модель .....	33
3.3 Аналіз результатів .....	38
Висновки до розділу 3 .....	39
Висновки .....	40
Перелік посилань .....	41

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕМ (expectation–maximization) - ітеративний метод пошуку оцінок максимальної схожості

MoG (Mixture of Gaussians) - модель гаусової суміші використовується для знаходження одно- або багатовимірних векторів невідомих параметрів нормальних розподілів.

$\Sigma$  - матриця коваріацій

$P(k)$  - апіорна ймовірність  $k$ -го класу.

$f()$  - функція щільності розподілу.

$\eta()$  - багатовимірна функція щільності розподілу.

$\alpha()$  - наближення до істинної апостеріорної ймовірності.

## ВСТУП

**Актуальність роботи.** Область використання відновлених параметрів фону лежить від стеження за рухомими об'єктами до видалення об'єктів з зображення відеозапису. Для розв'язання цієї задачі в загальному використовується два способи - класичний статистичний метод та штучні нейронні мережі. Задля максимальної надійності, математичної обґрунтованості та оскільки постановка задачі вимагає загального розв'язання, було прийняте рішення використовувати статистичні методи.

**Об'єкт дослідження** - статичне відеозображення з рухомими об'єктами.

**Предмет дослідження** - оцінки параметрів ймовірних моделей.

**Мета дослідження** - побудова алгоритму відновлення статистичних параметрів фону.

Завдання наступні:

- 1) Дослідити відомі методи для виділення фону
- 2) Обрати один з методів для подальшої роботи над ним.
- 3) Застосувати ЕМ алгоритм для розв'язання задачі.
- 4) Розробити програмну реалізацію алгоритму.
- 5) Дослідити розроблений алгоритм.

**Практичне значення одержаних результатів.** Відновлення статистичних параметрів фону використовується для стеження за об'єктами, деякі класичні алгоритми слідування сильно полягаються на маску об'єкта відтворену по цим параметрам. Відповідно якість цієї маски вимагається якомога краща. Було з'ясовано, що розроблений алгоритм в ряді випадків дає результати кращі, ніж вже реалізовані у відомих програмних засобах комп'ютерного зору.

**Наукова новизна.** Протягом роботи над дипломом було розроблено нову модель для виділення фону з відеозображення.

## **1 ПОПЕРЕДНІ РОБОТИ ПРИСВЯЧЕНІ ВІДНОВЛЕННЮ ПАРАМЕТРІВ ФОНУ ТА СЛІДКУВАННЮ ЗА ОБ'ЄКТАМИ**

У цьому розділі буде проаналізовано вже існуючі алгоритми відновлення фону.

Найпростішим способом отримання параметрів фону є отримання фонового зображення, яке не містить жодного рухомого об'єкту. Проте у деяких випадках весь фон не доступний одномоментно і має бути оновлений в критичних ситуаціях, таких як зміни освітлення, введення та виведення об'єктів зі сцени. Щоб врахувати ці випадки, було розроблено багато методів фонового моделювання. Ці методи моделювання можуть бути класифіковані за такими категоріями: базове фонове моделювання [2, 3], статистичне моделювання фону [1, 7], моделювання нечіткого фону [8, 9] та оцінка фону [10, 11].

### **1) Базове моделювання фону.**

У цьому випадку фон моделюється використовуючи середнє[2] або медіанне[5] значення інтенсивності пікселя в часі. Коли модель фону вважається знайденою, рішення про належність поточного стану пікселя до класу фону чи об'єкту приймається по його відхиленню.

### **2) Статистичне моделювання фону.**

У цьому випадку фон представляється у вигляді одного Гауссового, чи суміші Гауссових розподілів[1], або отримується через ядерну оцінку щільності [7].

### **3) Моделювання фону базоване на кластеризуванні.**

Цей підхід припускає, що кожне значення пікселя можна кластеризувати. Кожне нове значення пікселя відноситься до якогось з кластерів. А рішення, який з кластерів прийняти за фон приймається з якихось розумних для відповідного випадку ідей. Сам процес кластеризування здійснюється відомим методом K-means, або Codebook [4].

### **4) Моделювання фону нейронними мережами.**

Використання моделювання фону нейронними мережами дає кращі за статистичне моделювання результати у випадках неперіодичної зміни фону [6]. Задовільний результат було отримано використовуючи PNN (probabilistic neural network) та HNN (Hopfield neural network).

5) З використанням фільтрів.

Рішення про неналежність деякого значення пікселя до моделі фону приймається якщо воно значно відхиляється від передбаченої величини. Звичайними для такої задачі фільтрами являються фільтри Віннера, Калмана, Чебишева.

### 1.1 Базове моделювання фону

Різниця кадрів. Ідея методу полягає в обранні фоном попереднього кадру, і взяття його попиксельної різниці з поточним. Наявність різниці інтерпретується як рух об'єкту.

$$|I_t(x, y) - I_{t-1}(x, y)| > T \quad (1.1)$$

Перевагами методу є надвисока чутливість та простота алгоритму з константним часом виконання.

По середньому значенню. В цьому випадку за значення фону приймається середнє значення всіх попередніх станів пікселя.

$$B_t(x, y) = \frac{1}{N} \sum_{i=1}^{N-1} I_{t-i}(x, y) \quad (1.2)$$

$$|I_t(x, y) - B_t(x, y)| > T \quad (1.3)$$

Медіанне фільтрування належить до класу нерекурсивних алгоритмів. Робиться припущення, що фон буде медіаною для кожного пікселя зображення. Це припущення передбачає, що фон в кожному пікселі спостерігається більш ніж половину часу. Складність обрахунків для



медіанного фільтра складає  $N \log N$ , для кожного пікселя.

$$B_t(x, y) = \text{median}; I_{t-i}(x, y), \text{ where } i \in 0..n - 1 \quad (1.4)$$

$$|I_t(x, y) - B_{t-1}(x, y)| > T \quad (1.5)$$

$I_t(x, y)$  - значення інтенсивності пікселя  $(x, y)$  в момент часу  $t$

$B_{t-1}(x, y)$  - значення інтенсивності моделі фону цього пікселя в попередній момент часу

Константа  $T$  - поріг, значення якого зазвичай вибирається у відповідності з вимогами до чутливості.

## 1.2 Кластеризування

Кластеризування, або інакше сегментація зображення. Воно розділяється на класи, такі як методи базовані на знаходженні границь, порогові, регіонні, на нейронних мережах, кластеризаційні та гібридні. Тобто сегментація розділяє зображення на деяку кількість частин (зон), виходячи з таких їх властивостей як інтенсивність, колір, текстура, глибина та характер руху.

Сегментація на основі порогу є однією з найпростіших та найпотужніших технік. Порогова величина ділить зображення на два класи, все що менше відноситься до одного класу, все що більше до іншого. Головною задачею є спосіб вибору цієї порогової величини.[16]

Сегментація основана на виявленні країв зображення проводить пошук однотипних ділянок та виявляє зони різкої зміни їх параметрів, позначаючи їх як краї. В подальшому ділянки країв стягуються, щоб утворити замкнену криву, по якій походить остаточне розділення зображення. Такий метод має перевагу в можливості роботи не тільки на відеозображенні, а й на окремому фото.

Регіонні методи схожі до методів виявлення країв, проте при остаточному

розділенні зображення увага надається не замкненим краям, а належності конкретного пікселю до того чи іншого регіону по його параметрам.

### K-means алгоритм

Алгоритм кластеризування K-means це один з найпопулярніших методів метод розділення набору даних на визначену кількість груп.

K-means складається з двох етапів та належить до класу ітераційних алгоритмів.

На першому етапі він обраховує k-центроїд, а на другому відносить кожну точку до найближчого з k класів, базуючись на обрахованих до цього k центрах. Насправді є багато методів визначаючих відстань до класу (центру класу), проте найуживанішим залишається обрахування Евклідової відстані.

Коли обидва етапи завершені, повторюється перший етап визначення k-центроїду та обрахування нової відстані для кожної точки до цих k центрів і так далі. Кожен з центрів кластерів як точка з мінімальною сумарною відстанню до всіх членів кластеру. Тоді кожен кластер характеризується точками членами цього кластеру та його центром.

Тобто k-means алгоритм мінімізує суму відстаней від кожного члену кластеру до його центра, по всім кластерам.

Для випадку кластеризації для одного зображення алгоритм буде мати такі кроки[14]:

- 1) Ініціалізувати кількість кластерів та їх центри.
- 2) Для кожного пікселя обрахувати його Евклідову відстань  $d$  до кожного центру по формулі нижче.

$$d_k(x, y) = ||p(x, y) - c_k|| \quad (1.6)$$

- 3) Віднести кожен піксель до деякого класу, вибираючи найменшу з його відстаней  $d$ .

- 4) Після того, як всі пікселі кластеризовані, обрахувати нові координати центрів.

$$c_k = \frac{1}{|k|} \sum_{y \in c_k} \sum_{x \in c_k} p(x, y) \quad (1.7)$$

5) Повторювати кроки 2-4, допоки алгоритм не досягне умови зупинки.

$p(x, y)$  - координата (значення інтенсивності пікселя) в точці  $x, y$ .

$c_k$  - координата (інтенсивність) центра кластеру  $k$ .

$|k|$  - кількість пікселів, віднесених до кластеру  $k$ .

$d_k(x, y)$  - відстань від точки  $x, y$  до центру з номером  $k$ .

Таким чином, отримаємо кластеризацію зображення на  $k$  класів, як це показано в прикладі.

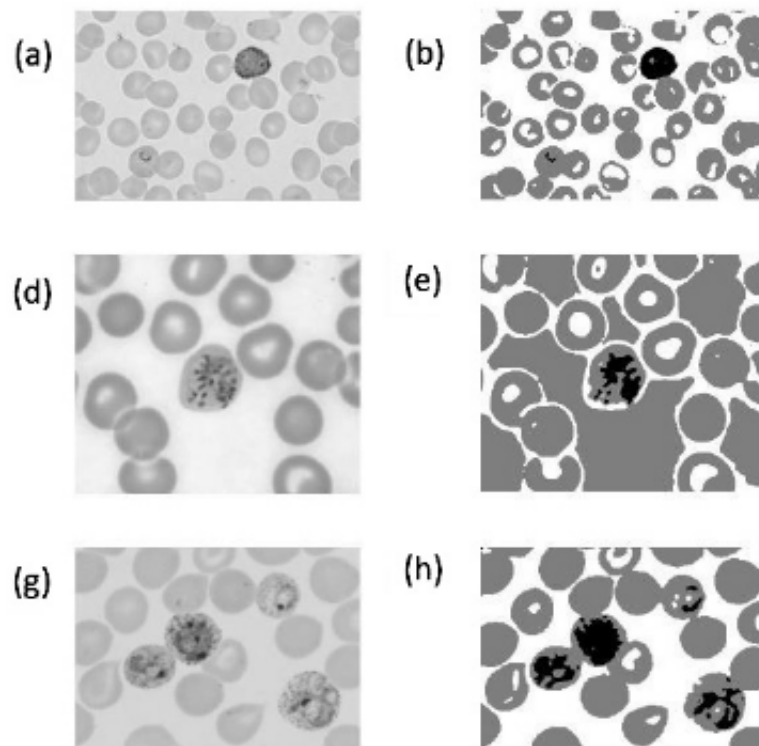


Рисунок 1.1 – Результат роботи алгоритму k-means

На цьому рисунку a,d,g це вихідні зображення, а b,e,h - результат роботи k-means. Чорним та сірим кольором виділено два різних класи.

Вище наведений алгоритм працює окремо на кожному зображенні у відео потоці. Проте має модифікацію для роботи над кожним пікселем окремо, використовуючи його послідовності станів в часі:

- 1) Ініціалізувати кількість кластерів та їх центри.
- 2) Для кожного значення пікселя обрахувати його Евклідову відстань  $d$

до кожного центру по формулі нижче.

$$d_k^i = ||p^i - c_k|| \quad (1.8)$$

3) Віднести кожне значення пікселя до деякого класу, вибираючи найменшу з його відстаней  $d$ .

4) Після того, як всі значення кластеризовані, обрахувати нові координати центрів.

$$c_k = \frac{1}{|k|} \sum_{i \in c_k} p^i \quad (1.9)$$

5) Повторювати кроки 2-4, допоки алгоритм не досягне умови зупинки.

Умовою припинення роботи алгоритму вибирають величину відхилення координат центра відносно його попереднього положення.

$$||c_k^t - c_k^{t-1}|| < E \quad (1.10)$$

Де  $E$  деяка задана допустима величина помилки.

Codebook алгоритм [15].

У цьому методі кожен піксель представляється кодовою книгою  $C = c_1, c_2, \dots, c_L$ , довжина кодової книги, в загальному, відрізняється для кожного з пікселів. Кожне кодове слово  $c_i, i = 1, \dots, L$  отримується з RGB вектору  $v_i = (R_i, G_i, B_i)$  та шістки  $aux_i = (I_i^{min}, I_i^{max}, f_i, p_i, \lambda_i, q_i)$  де  $I_i^{min}$  та  $I_i^{max}$  відповідно мінімальна та максимальна інтенсивність всіх пікселів віднесених до кодового слова  $c_i$ .  $f_i$  - це частота з'явлень відповідного кодового слова,  $\lambda_i$  - це найдовший інтервал часу навчання, що кодове відповідне слово не повторювалось.  $p_i$  та  $q_i$  - перша та остання зустріч кодового слова.

Кодова книга створюється або оновлюється, по двом критеріям. Перший критерій заснований на дисперсії кольору, а другий на дисперсії яскравості. Дисперсія кольору обраховується згідно формули:

$$\delta = \sqrt{||p_t||^2 - C_p^2} \quad (1.11)$$

$C_p^2$  коефіцієнт автокореляції R, G та B каналів кольору пікселя та кодового слова  $c_i$ , нормалізованих по яскравості.

$$C_p^2 = \frac{(R_i R + G_i G + B_i B)^2}{R_i^2 + G_i^2 + B_i^2} \quad (1.12)$$

Яскравість для пікселя обраховується по наступній формулі.

$$I = \sqrt{R^2 + G^2 + B^2} \quad (1.13)$$

Та обмежується двома границями.

$$I_{low} = I_i^{max} \quad I_{hi} = \min(\beta I_i^{max}, \frac{I_i^{min}}{\alpha}) \quad (1.14)$$

Для кожного пікселя знаходиться задовольняюче його за обома умовами кодове слово та оновлюється наступним чином:

$$v_i = (\frac{f_i R_i + R}{f_i + 1}, \frac{f_i G_i + G}{f_i + 1}, \frac{f_i B_i + B}{f_i + 1}) \quad (1.15)$$

$$aux_L = (\min(I, I_i^{min}), \max(I, I_i^{max}), f_i + 1, \max(\lambda_i, t - q_i), pi, t) \quad (1.16)$$

Якщо задовольняюче кодове слово не знайдене, створюється нове. Тоді:

$$v_t = (R, G, B), \quad aux_K = (I, I, 1, t - 1, t, t) \quad (1.17)$$

Після періоду навчання, якщо піксель не відноситься ні до одного кодового слова, то він поміщається в "кеш" слово та помічається як фон, якщо "кеш" слово зустрічається досить часто, то воно поміщається в кодову книгу.

### 1.3 Статистичі методи

MoG Суміш Гауссових розподілів - MoG, є відомим методом для відновлення фону та виявлення рухомих об'єктів з статичних камер. Протягом останніх років було запропоновано численні вдосконалення оригінального методу, розробленого Стауфаером та Грімсоном [1].

Кожен піксель характеризується інтенсивністю в RGB просторі. Тоді ймовірність спостереження пікселя з поточними характеристиками інтенсивності описується наступним чином:

$$P(X_t) = \sum_{i=1}^K w_{i,t} \cdot \eta(X_t, m_{i,t}, \Sigma_{i,t}) \quad (1.18)$$

$X_t$  - трьох вимірний параметр інтенсивності пікселя, в момент часу  $t$ .  $w_{i,t}$  - деякі параметри ваги, асоційовані з відповідним  $i$ -тим Гауссіаном в момент часу  $t$ .

$\eta$  - функція розподілу ймовірностей.

$m_{i,t}, \Sigma_{i,t}$  - параметри моделі.

Сама ж функція розподілу ймовірностей для багатомірного випадку має вигляд:

$$\eta(X_t, m, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t-m)^T \Sigma^{-1} (X_t-m)} \quad (1.19)$$

Задля простоти обрахувань Грімсон та Стауфер припустили, що матриця коваріації складається з незалежних величин, тобто:

$$\Sigma_{i,t} = \sigma_{i,t}^2 I \quad (1.20)$$

Тоді для трьох вимірний випадку RGB Гауссіани приймають вигляд як на Рисунок 1.1

Для встановлення параметрів моделі MoG класично застосовується ЕМ

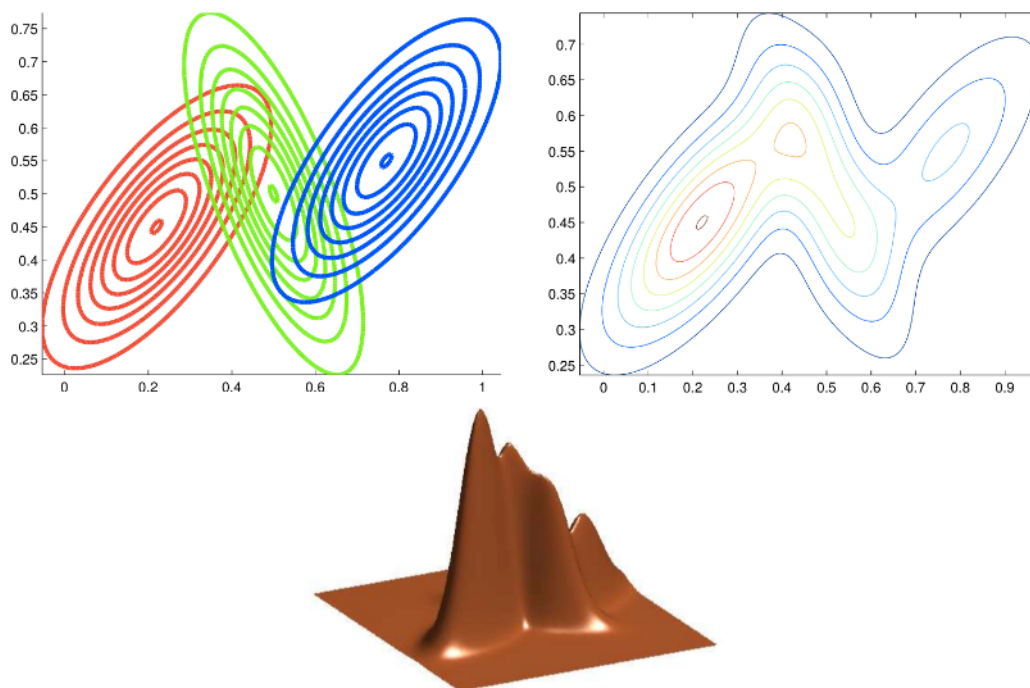


Рисунок 1.2 – Візуалізація мікстури Гауссіан

алгоритм, як засіб кластерного аналізу.

ЕМ-алгоритм обрано з міркувань, що в моделі випадковими являються як стани  $K$ , тобто належить піксель об'єкту чи фону, так і самі спостереження  $X_t$  - інтенсивність пікселя в момент часу  $t$ . Найважливішим уточненням є те, що задача не належить класу Байєсовських, адже хоч спостереження  $X$  відомі, проте умовні ймовірності  $P(X|k)$  спостережень  $X$  при умові, що об'єкт знаходиться в стані  $k$  існують, але не відомі,  $X$  та  $k$  залишаються випадковими, а апіорна ймовірність  $P(k)$  також хоча і існує, проте невідома.

Таким чином задача знаходження параметрів моделі MoG належить класу задач кластеризації.

ЕМ алгоритм забезпечує ітеративне знаходження оцінок максимальної правдоподібності параметрів ймовірнісних моделей. Кожна ітерація складається з двох кроків:

- 1) На Е-кроці вираховується очікуване значення вектора прихованих змінних, по поточному наближенні векторів параметрів обраної моделі.
- 2) На М-кроці вирішується задача максимізації правдоподібності та

знаходиться наступне наближення векторів параметрів моделі.

Логарифм правдоподібності розписується так:

$$\ln P(X|w, \eta, \Sigma) = \sum_{n=1}^N \ln \left( \sum_{k=1}^K w_k \eta(x_n | m_k, \Sigma_k) \right) \quad (1.21)$$

Такого вигляду рівняння дуже важко максимізувати, тому вводиться додаткова змінна:

$$l(m) = \prod_{i=1}^N P(x_i, k_i) = \prod_{i=1}^N P(k_i) P(x_i | k_i) = \prod_{i=1}^N P(k_i) \prod_{i=1}^N P(x_i | k_i) \quad (1.22)$$

де  $P(x_i | k_i) = f(x_i, a_{k_i})$ , та нехай  $L(m) = \ln l(m)$

$$L(m) = \sum_{i=1}^N \ln p_{k_i} + \sum_{i=1}^N \ln f(x_i, a_{k_i}) \quad (1.23)$$

Тоді для таких змінних  $\alpha(i, k)$ , що:

$$\sum_{k \in K} \alpha(i, k) = 1 \quad (1.24)$$

справедливо наступне:

$$L(m) = \sum_{n=1}^N \sum_{k \in K} \alpha(i, k) \ln P(k) + \sum_{i=1}^N \sum_{k \in K} \alpha(i, k) \ln f(x_i, a_k) \quad (1.25)$$

А отже, максимізацію правдоподібності можливо проводити по окремості.

З першої частини отримуємо:

$$\begin{cases} P(k) = \arg \max (\sum_{i=1}^N \alpha(i, k) \ln P(k)) \\ \sum_{k \in K} P(k) = 1 \end{cases} \Rightarrow \max P(k) = \frac{\sum_{i=1}^N \alpha(i, k)}{N} \quad (1.26)$$

З другої частини для задачі пошуку параметрів МоГ в одновимірному випадку (або за гіпотези незалежності), приймаючи  $f(x_i, a_k)$  як функцію



щільності розподілу ймовірності Гауссового розподілу одержуємо:

$$m_k^* = \arg \max_m \sum_{i=1}^N \alpha(i, k) \ln \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_i - m)} = \arg \min_m \sum_{i=1}^N \alpha(i, k) (x_i - m)^2 \Rightarrow \quad (1.27)$$

$$m_k^* = \frac{\sum_{i=1}^N \alpha(i, k) x_i}{\sum_{i=1}^N \alpha(i, k)}, d_k^* = \sqrt{\frac{\sum_{i=1}^N \alpha(i, k) (x_i - m_k^*)^2}{\sum_{i=1}^N \alpha(i, k)}} \quad (1.28)$$

А  $\alpha(i, k)$  визначається, за умови, що це наближення до апостеріорної ймовірності, що спостереження  $X_i$  належить класу  $k$ .

$$\alpha(i, k) = \frac{P(k) f(x_i, a_k)}{\sum_{k \in K} P(k) f(x_i, a_k)} \quad (1.29)$$

Для багатовимірного випадку це відповідно:

$$\bar{m}_k = \frac{\sum_{i=1}^N \bar{\alpha}(i, k) \bar{x}^i}{\sum_{i=1}^N \bar{\alpha}(i, k)} \quad (1.30)$$

$$\bar{d}_k = \frac{\sum_{i=1}^N \bar{\alpha}(i, k) (\bar{x}^i - \bar{m}_k)^T (\bar{x}^i - \bar{m}_k)}{\sum_{i=1}^N \bar{\alpha}(i, k)} \quad (1.31)$$

Приклад роботи ЕМ на двовимірному просторі

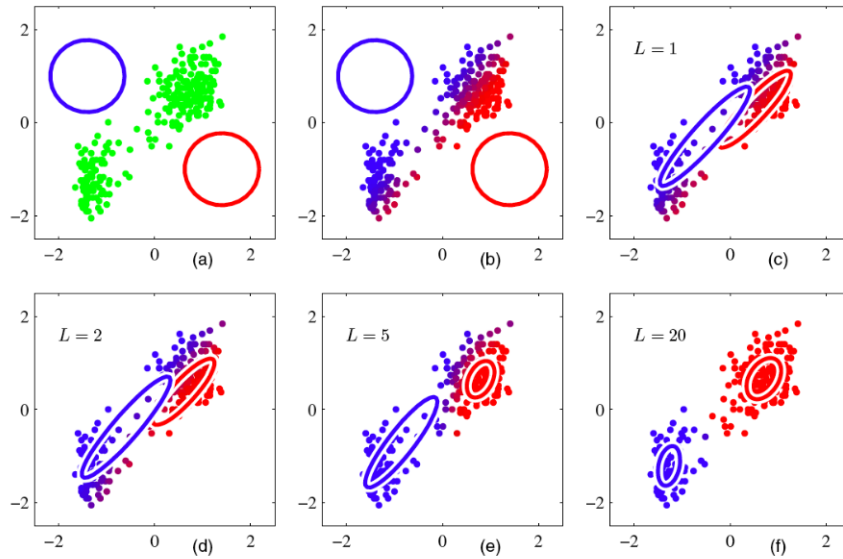


Рисунок 1.3 – Ітерації роботи ЕМ

Іншим потужним методом статистичного аналізу є KDE - Kernel Density Estimation. Показано, що оцінка щільності ядра асимптотично сходиться до будь-якої функції щільності. Це робить цей метод досить загальним та застосовним до багатьох задач з розпізнавання образів. Загальний опис:

Дано вибірку  $S = x_{i=1...N}$  з розподілу з деякою невідомою функцією щільності  $p(x)$ , оцінка цієї функції щільності може бути визначена як:

$$\hat{p}(x_t) = \frac{1}{N} \sum_{i=1}^N K_{\sigma}(x - x_i) \quad (1.32)$$

$K_{\sigma}$  - ядерна функція (вікно).  $\sigma$  - ширина вікна. Ядро - вагова функція для оцінки параметрів і розподілів. Ядерна функція має задовільняти такі умови:

$$\int_{-\infty}^{+\infty} K(u) du = 1 \quad (1.33)$$

$$K(-u) = K(u) \quad K(u) \geq 0 \quad (1.34)$$

Така оцінка може бути розширена до випадку багатьох розподілів:

$$\hat{p}(x_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^3 K_{\sigma_j}(x_{t_j} - x_{i_j}) \quad (1.35)$$

Тут  $x_t$  - тривектор з даними про R, G та B складові.

Якщо ми оберемо нашу ядерну функцію Гауссіаном, то отримаємо такий вигляд:

$$\hat{p}(x_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^3 \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x_{t_j}-x_{i_j})^2}{2\sigma_j^2}} \quad (1.36)$$

Піксель позначається як об'єкт, якщо задовольняє умові:

$$\hat{p}(x_t) < th \quad (1.37)$$

Приймаючи стан пікселя за об'єкт, якщо  $\hat{p}(x_t) < th$ , де  $th$  деякий глобальний поріг для всіх зображень, отримаємо розділення відео на

фон-об'єкт.

### **Висновки до розділу 1**

У цьому розділі було детально наведено відомі методи розділення та кластеризації зображення, а також пошуку статистичних параметрів. А також описаний загальний підхід до проблеми.

## 2 АНАЛІЗ РОБОТИ ВІДОМИХ АЛГОРИТМІВ

У цьому розділі буде проаналізована робота алгоритмів для визначення параметрів фону та виділення об'єктів.

### 2.1 Базове моделювання фону

Різниця кадрів.

Метод прийняття за фон попереднього кадру та взяття попиксельної різниці між ним та поточним кадром хоча і являється чутливим до руху методом, проте дає тільки інформацію про зміну чогось в відеопотоці, будь то зміна яскравості освітлення чи рух об'єкта. Приклад роботи приведений на наступному рисунку.[17]

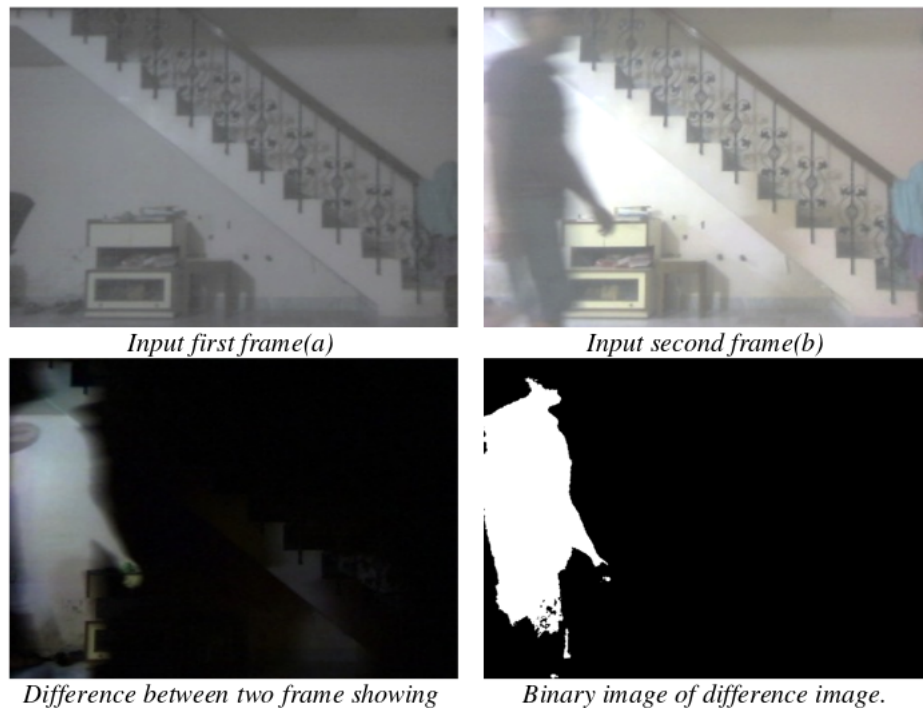


Рисунок 2.1 – Робота алгоритму різниці кадрів

Таким чином, виділення зображення фону з відеопотоку цим методом є досить складною задачею.

Одним з прийнятних методів відновлення фону через алгоритм взяття різниці кадрів могла б бути така реалізація:

- 1) Для кожного пікселя розрахувати різницю  $d$  з його попереднім станом:

$$d_t(x, y) = |p_t(x, y) - p_{t-1}(x, y)| \text{ for } \forall x, y, t = 1, \dots, N \quad (2.1)$$

- 2) По розрахованій різниці провести порівняння з деякою пороговою величиною  $th$  та розмітити об'єкт - фон:

$$\text{if } d_t(x, y) > th \text{ then } L_b(x, y, t) = 1, \text{ else } L_b(x, y, t) = 0; \quad (2.2)$$

- 3) Провести розрахунок параметрів фону  $b(x, y)$  по всім пікселям, що визначені фоном:

$$b(x, y) = \frac{1}{|L_b(x, y)|} \sum_{t=1}^N p_t(x, y) 1(L_b(x, y, t) = 1) \quad (2.3)$$

Отриманий результат інтенсивності фону буде середнім значенням всіх пікселів, що не класифікувались як об'єкт. Отриманий результат буде без будь-яких гарантій на коректність, адже не рухаючийся об'єкт моментально відноситься до фону.

Таким чином, різниця кадрів, хоча і являється потужним методом виявлення руху, проте абсолютно не застосовний для відновлення фону відеозображення.

Відновлення фону за середнім значенням буде давати навіть гірші результати, ніж відновлення фону алгоритмом знаходження різниці кадрів. Це досить очевидно з формули розрахунку параметрів фону.

$$B_t(x, y) = \frac{1}{N} \sum_{i=1}^{N-1} I_{t-i}(x, y) \quad (2.4)$$

Медіанний фільтр. Більш розумним підходом до відновлення фону є

медіанний фільтр. Цей метод дає чітке розділення на класи фон - об'єкт. Проте він робить припущення про те, що фон з'являється в кожному пікселі більше ніж половину часу. Це працює з допустимою якістю для випадків камери з незавантаженої дороги:

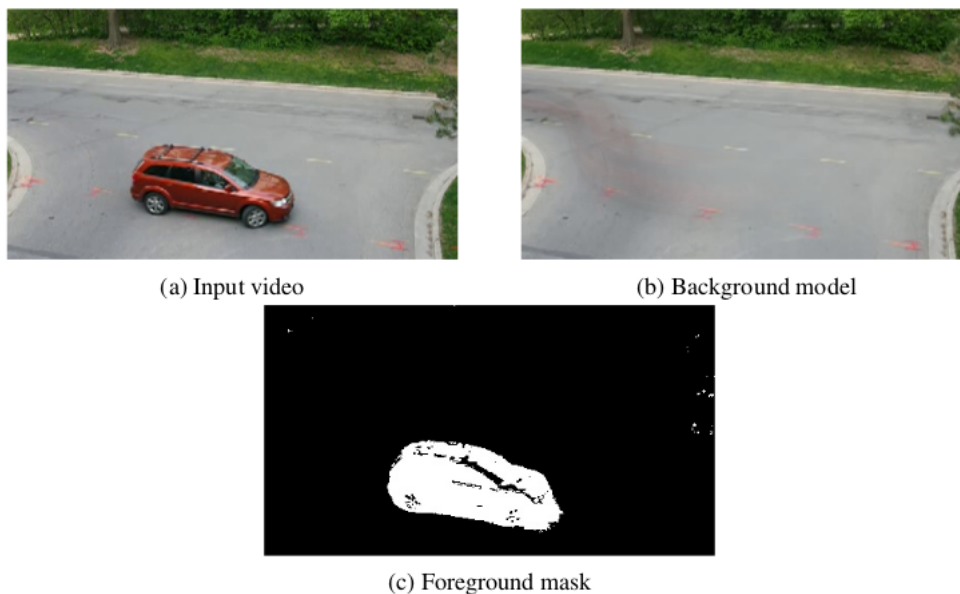


Рисунок 2.2 – Відновлений фон медіанним фільтром та маска

Проте у випадках відносно статичного об'єкту як, наприклад, лектор перед дошкою, цей метод місцями класифікує лектора як фон.

Отже базове моделювання фону не буде давати гарні результати у таких випадках:

- 1) Якщо на відео присутні об'єкти, що повільно рухаються.
- 2) Коли присутня відносно різка зміна освітлення.
- 3) Якщо істинний фон не являється однотипним.

Ще одним важливим моментом є те, що всі вищенаведені методи використовують деяку порогову величину  $th$ , визначену для всього зображення та всіх моментів часу одночасно. Проблема в тому, що необхідна величина  $th$  для різних ділянок зображення різна, тобто в яскравому освітленні різниця між об'єктом та фоном буде більш помітна, ніж при темному.

TH should be different for each pixel

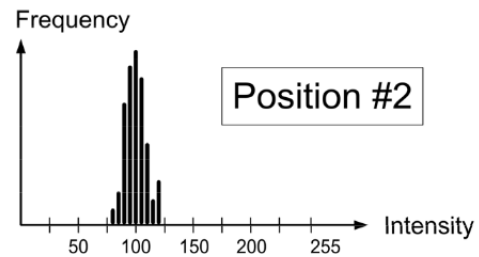
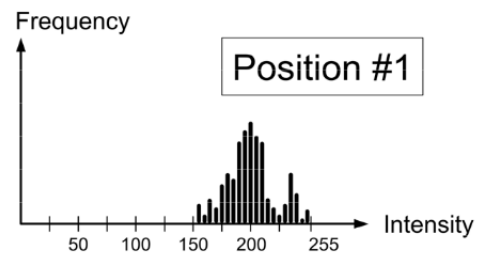


Рисунок 2.3 – Залежність значення порогової величини від освітлення

Частково ця проблема вирішується заміною виразу для порівняння:

$$|I(x, y, y) - B(x, y, t)| > th \quad (2.5)$$

на

$$\frac{|I(x, y, y) - B(x, y, t)|}{B(x, y, t)} > th \quad (2.6)$$

## 2.2 Моделювання фону базоване на кластеризуванні

### K-means

Алгоритм k-means застосований до всього кадру зображення дає результати наступного вигляду: Таке розділення дає змогу розбити зображення на класи та потенційно їх розділити на об'єкт-фон. Проте, в загальному алгоритм не надає ніякої інформації про належність деякого класу до фону чи об'єкту.

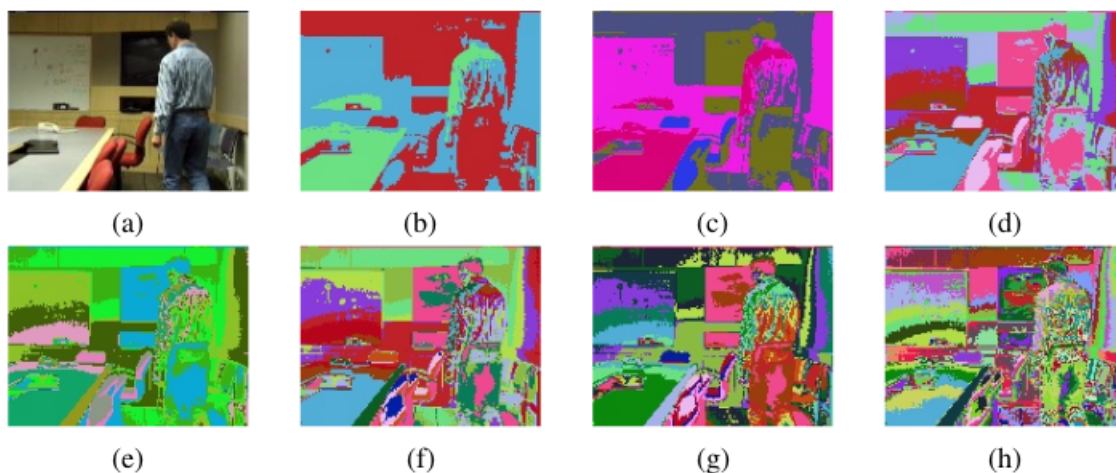


Рисунок 2.4 – зображення b-h кластеризування на 3-5-8-10-15-20-50 кластерів відповідно

Та ж сама ситуація і з роботою цього алгоритму для значень кожного пікселя по окремості.

#### Codebook

На відміну від k-means, codebook не тільки кластеризує, а й дає класифікацію фон-об'єкт для кожного пікселя.

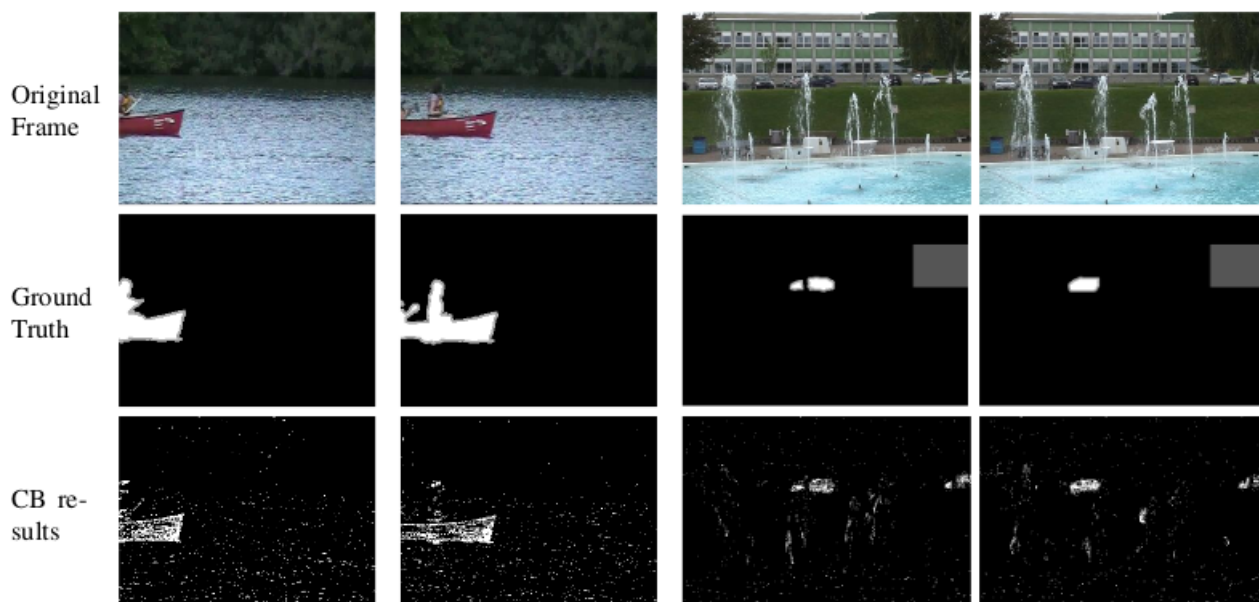


Рисунок 2.5 – Codebook алгоритм

Таким чином до об'єкту відноситься кожен піксель, що не відповідає ні



одному кодовому слову.

$$BS\_result = \begin{cases} background & \text{if } brightness(I, \langle I_i^{min}, I_i^{max} \rangle) \cap dist(x_t, v_i) < \sigma \\ foreground & \text{otherwise} \end{cases} \quad (2.7)$$

Проте цей алгоритм має декілька важливих недоліків:[18]

1) В умовах високого контрасту зображення, наприклад в зимній період часу, деякі частини чорного одягу можуть віднеситись до кодового слова темного асфальту, не дивлячись на те, що вони досить сильно різняться, і таким чином класифікуватись як фон.

2) Алгоритм передбачає визначення деяких параметрів  $\alpha$  та  $\beta$  людиною. Більше того, вони будуть специфічними для кожного відео.

### 2.3 Статистичні методи відновлення фону

Алгоритм пошуку об'єктів на статичному фоні реалізований у бібліотеці `opencv`, а саме `MOG` та `MOG2`.

Приклад роботи `MoG`:

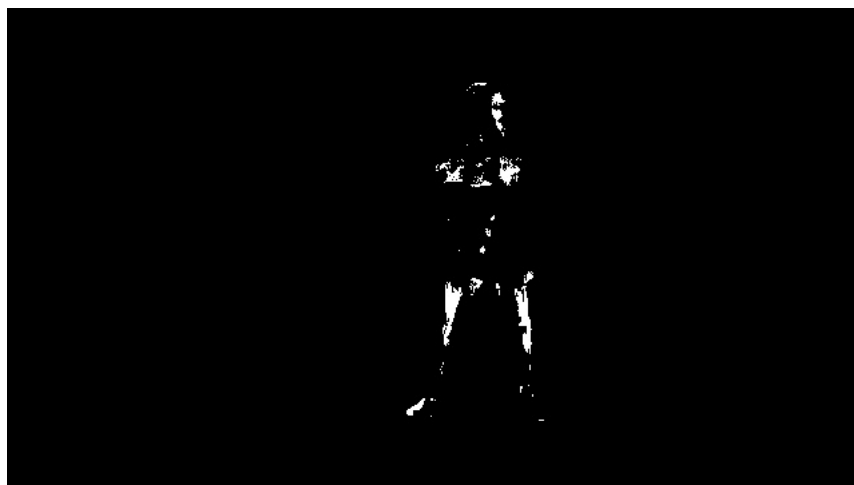


Рисунок 2.6 – Приклад роботи `MOG` в `opencv`

На рисунку 2.6 білим кольором позначено області, що визначені алгоритмом як об'єкт.

Головна проблема в тому, що об'єкт хоча і не статичний, проте в сумі більше половини часу проводить на одному місці.

Краще з такими ситуаціями справляється адаптований алгоритм MoG2, проте він має схильність виділяти і об'єкт і фон в один клас, що також виражається в чорних плямах на об'єкті, Рисунок 2.7.



Рисунок 2.7 – Приклад роботи MOG2 в opencv

Не дивлячись на неякісний кінцевий результат, розділення на класи в MoG було виконано, проте кінцева класифікація на фон-об'єкт спрацювала неправильно, і цьому є декілька причин.

Для наглядного показу проблеми було самостійно написано алгоритм ЕМ для знаходження параметрів моделі MoG.

Проте як відомо отримання параметрів гауссових розподілів ще нічого не каже про те належить він фонові, чи об'єкту.

На рисунку 2.8, червоним позначено значення пікселів, що віднесені до класу об'єкту, а синім до фону.

Опираючись на емпіричні данні, представлені на рисунку 2.8 заключимо, що фоном являється клас з меншою дисперсією  $D_f < D_o$ . Класифікуючи тільки по цьому признаку, отримали результати, представлені на рисунку 2.9.

Не звертаючи увагу на викиди у вигляді білого шуму по сторонам, не

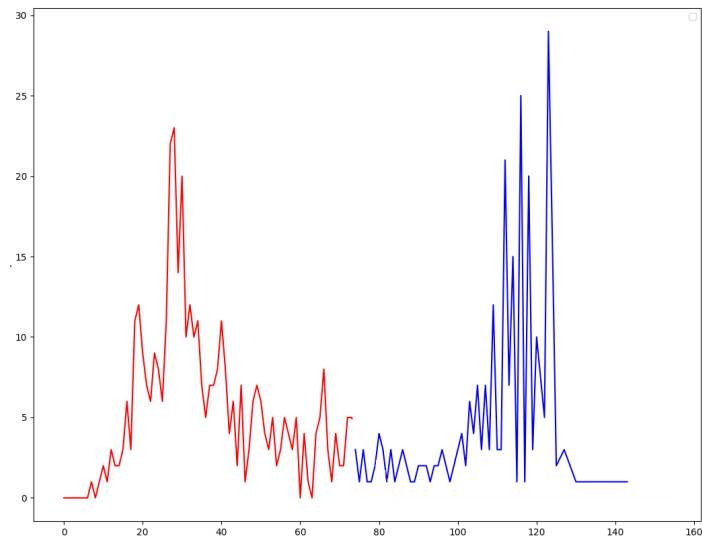


Рисунок 2.8 – Розбиття на класи

дуже динамічний об'єкт виділився краще, проте залишилася велика біла пляма в центрі.



Рисунок 2.9 – Рішення про класи по дисперсії

Аналізуючи його видно, що класифікатор відпрацював так, що клас з меншою дисперсією виявився об'єктом. Це спричинено двома обставинами:

- 1) об'єкт й справді був досить монотонним в тій точці
- 2) багаточисельні викиди змусили один з класів (фону) суттєво збільшити свою дисперсію. Викиди, в основному спричинені тінню.

Іншою критичною для такого підходу ситуацією став такий розподіл значень пікселя - рисунок 2.10.

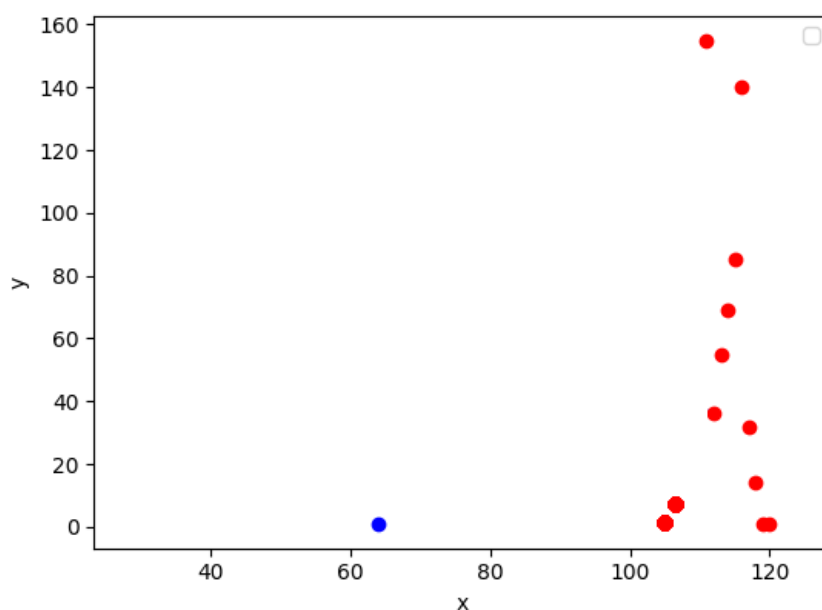


Рисунок 2.10 – Критична ситуація розподілу значень зображення

Розподіли знайдені коректно, проте спостереження зі значенням інтенсивності пікселя 64 класифікувалось в окремий клас з нульовою дисперсією. Така ситуація з'являється в областях зображення, де об'єкт опиняється лише на 1-2 кадри, а фон змінює свій колір під впливом шумів на матриці відеокамери та змін освітлення сцени. Таким чином, об'єкт класифікувався як фон, а фон як об'єкт. Яскравим прикладом такої ситуації є проліт пташки в небі. На відновленому зображенні фону, по її траєкторії польоту залишається слід з чорних цяток.



Рисунок 2.11 – Проліт пташки

Третій випадок некоректної поведінки MoG - рисунок 2.12

Тут представлений випадок коли об'єкт не з'являвся в області пікселя

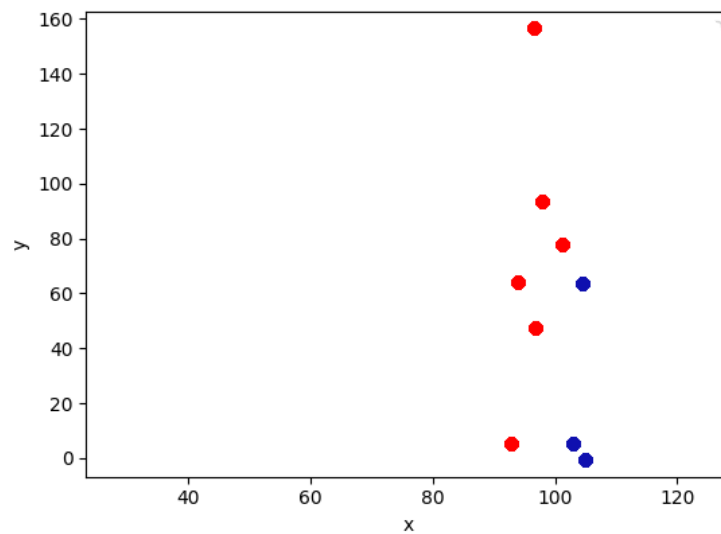


Рисунок 2.12 – Критична ситуація розподілу значень зображення

взагалі. Деякі флуктуації присутні з наведених вище причин. В силу своєї постановки алгоритм зобов'язаний знайти параметри для обох розподілів, тому в такому випадку фонове зображення також інтерпретується як таке, що містить об'єкт.

## Висновки до розділу 2

У цьому розділі було проаналізовано причину недостатньої якості роботи класичних алгоритмів та MoG. А також розглянуто одну з базових гіпотез, щодо правила розділення на класи фон-об'єкт.

### **3 РОЗРОБКА АЛГОРИТМУ ВІДНОВЛЕННЯ СТАТИСТИЧНИХ ПАРАМЕТРІВ ФОНУ**

У цьому розділі буде запропонований алгоритм для встановлення параметрів фону, який базується на проаналізованому у попередньому розділі алгоритмі.

#### **3.1 Підсумки з проаналізованих відомих алгоритмів**

З минулого розділу можна заключити що, на відміну від k-means, алгоритм має давати не тільки розбиття на класи, а й чітке указання на належність до класу фону та, на відміну від базових методів, має бути стійким до повільно рухаючихся об'єктів.

Ціллю для покращення була обрана модель MoG. Так, у попередньому розділі був отриманий результат, що значною мірою не вдається відновити параметри фону цією моделлю, по таким причинам:

- 1) Випадкові викиди, або схожі до фону значення інтенсивності об'єкту, "відтягують" один з розподілів на себе.
- 2) Некоректно кластеризується випадок без жодного спостереження об'єкту.
- 3) Некоректно інтерпретується випадок з всього лиш одним спостереженням об'єкту.
- 4) Алгоритми не стійкі до повільно рухаючихся об'єктів.

### 3.2 Запропонована модель

Передбачається, що вибірка:

$$X^{(n)} = \{X_1, \dots, X_n\} \quad (3.1)$$

$$X_j = \{x_{j1}, \dots, x_{jp}\}, \quad p \geq 1 \quad (3.2)$$

представляє собою реалізацію деякої неперервної  $k$ -мірної випадкової величини  $\eta = (\eta_1, \dots, \eta_k)$  з щільністю ймовірності  $f(X, \theta)$ , яка з достатньою точністю апроксимується щільністю ймовірності такої Гаусової суміші:

$$f(x, \theta) = (2 * \pi)^{\left(\frac{-p}{2}\right)} \sum_{i=1}^k \pi_i |\sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\eta_i)\sigma_i^{-1}(X-\eta_i)^T} \quad (3.3)$$

$$\theta = (\eta_1, \dots, \eta_k, \sigma_1, \dots, \sigma_k, \pi_1, \dots, \pi_k, k) \quad k \geq 2, \pi_i \geq 0, \sum_{i=1}^k \pi_i = 1 \quad (3.4)$$

Необхідно знайти оптимальні оцінки невідомих параметрів гаусової суміші -  $\theta$ .

Для визначення кількості параметрів суміші використовується багато відомих способів, таких як метод максимуму правдоподібності, мінімуму  $\chi^2$ , найменших квадратів і тому подібні.

Для нашої задачі було вирішено не застосовувати додаткових методів попередньої оцінки кількості класів суміші, а припустити що фоном є тільки один клас, а все інше прийняти за об'єкти.

Модель МоГ базується на гіпотезі, що значення фону та об'єкту мають Гаусовий розподіл.

Для перевірки гіпотези про належність вибірки значень фону нормальному розподілу можна застосувати критерій узгодженості Пірсона -

$\chi^2$ . Тобто перевірити гіпотезу:

$$H_0 : F_n \in \{F(x, \theta), \theta \in \theta\} \quad (3.5)$$

$F_n(x)$  - відома вибірка.

$H_0$  - гіпотеза про належність вибірки до перевіряймого розподілу

$\theta$  - вектор параметрів розподілу  $F(x, \theta)$

З датасету розмічених відеозображень, тобто з наявними масками об'єктів, було отримано тестові вибірки значень інтенсивностей пікселів фону.

Згідно з критерієм узгодженості Пірсона для кількох типів розподілу було розраховано статистику критерію узгодженості з кількістю інтервалів розбиття  $k = 100$ , від 0 до 255.

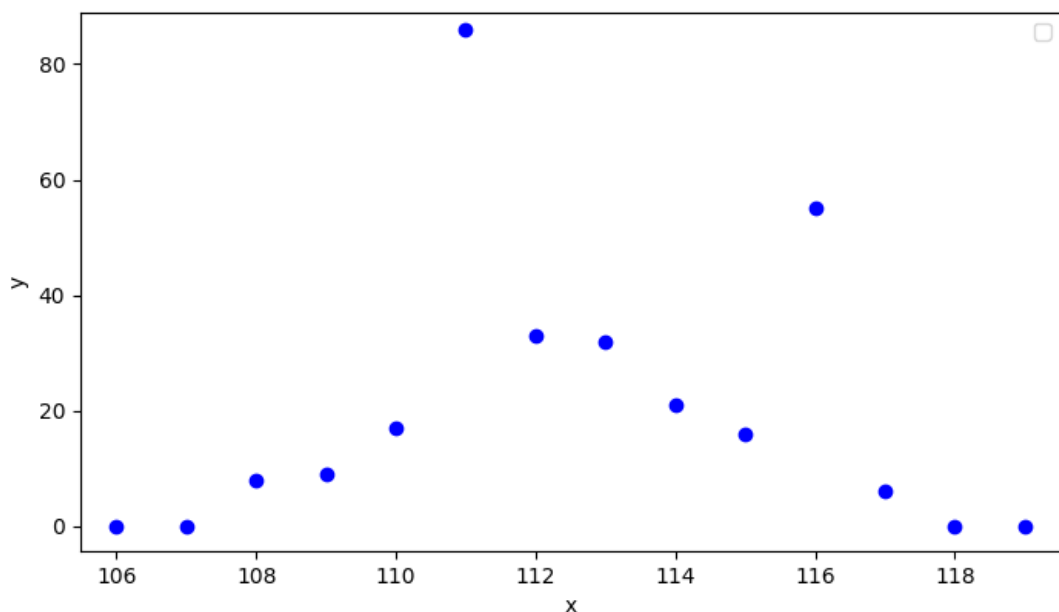


Рисунок 3.1 – Приклад вигляду вибірки значень фону

На рисунку 3.1 по горизонталі відмічено інтенсивність, а по вертикалі частота зустріч.

Розрахунки показали, що в усіх випадках розподіл вибірки узгоджується з Пуассоновим на нормальним розподілом, та ніколи не приймав вигляд рівномірного розподілу.



Тобто на відео зображенні фон буде приймати вигляд гаусового розподілу.

Натомість тестові вибірки інтенсивностей пікселів об'єкту хоча й приймають в основному вигляд нормального та Пуассонового розподілу, проте у 12% випадків підчиняються рівномірному розподілу. Приклад такої поведінки:

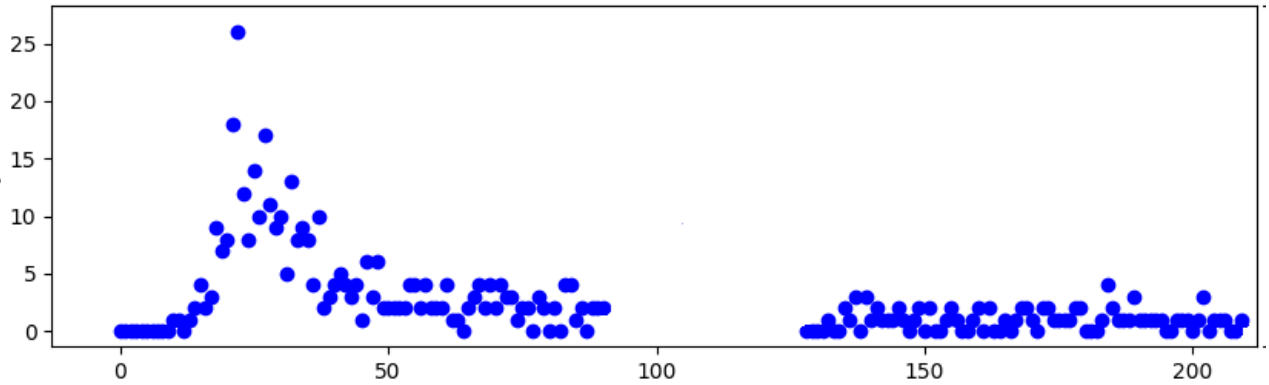


Рисунок 3.2 – Приклад вигляду вибірки значень об'єкта

Так, було запропоновано ввести в модель замість двох Гауссових класів один нормальний та один рівномірний розподіл, тоді для максимізації рівняння 1.25:

$$L(m) = \sum_{n=1}^N \sum_{k \in K} \alpha(i, k) \ln P(k) + \sum_{i=1}^N \sum_{k \in K} \alpha(i, k) \ln f(x_i, a_k) \quad (3.6)$$

застосовується наступний вигляд членів:

$$f_g(x) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{(x-m_g)^2}{2\sigma_g^2}} \quad f_u(x) = \frac{1}{b-a} \quad (3.7)$$

$$m_g^* = \frac{\sum_{i=1}^N \alpha(i, g) x_i}{\sum_{i=1}^N \alpha(i, g)} \quad d_g^* = \sqrt{\frac{\sum_{i=1}^N \alpha(i, g) (x_i - m_g)^2}{\sum_{i=1}^N \alpha(i, g)}} \quad (3.8)$$

$$m_u^* = \frac{a+b}{2} \quad d_u^* = \frac{(b-a)^2}{12} \quad (3.9)$$

$$\alpha(i, k) = \frac{P(k) f(x_i, a_k)}{\sum_{k \in K} P(k) f(x_i, a_k)} \quad (3.10)$$

Де  $m_g$  та  $d_g$  - математичне очікування та дисперсія для гаусової компоненти, а  $m_u$  та  $d_u$  - відповідні компоненти рівномірного розподілу. Параметри  $a$  та  $b$  для рівномірного розподілу в загальному вигляді визначаються як  $a = \min_x x$   $b = \max_x x$ , але задля більш стійкої до викидів роботи, покладемо їх як мінімально та максимально можливі відповідно, тобто 0 та 255.

Специфіка роботи алгоритму ЕМ вимагає указання деяких початкових значень для вектору параметрів  $\theta$ . Відомим фактом являється те, що ЕМ дуже чутливий до вказання початкових даних. Так якщо вони співпадуть на початку, то алгоритм просто не почне свою роботу. А також вони можуть співпасти впродовж самої роботи алгоритму, в такому випадку загальною рекомендацією є обрання випадкового параметра та штучне змінення його значень на не велику величину.

Проте протягом тестування розробленої програми було виявлено ще одну особливість роботи ЕМ для запропонованої моделі, а саме, результуючі знайдені параметри не сильно відхиляються від початкових заданих. Тобто при умові наявності в області деякої кількості значень пікселів, з великою ймовірністю результуюче математичне очікування буде знайдено в області близько 30 значень від початкових. Таким чином клас фону знаходиться не вірно.

Такі проблеми вирішують, здебільшого, введенням параметру швидкості навчання.

Для цієї роботи проблема була вирішена попереднім застосуванням класичного MoG для грубої оцінки центрів нормальних розподілів, а потім послідовним запуском модифікованого алгоритму з кожним знайденим центром в якості початкових даних.

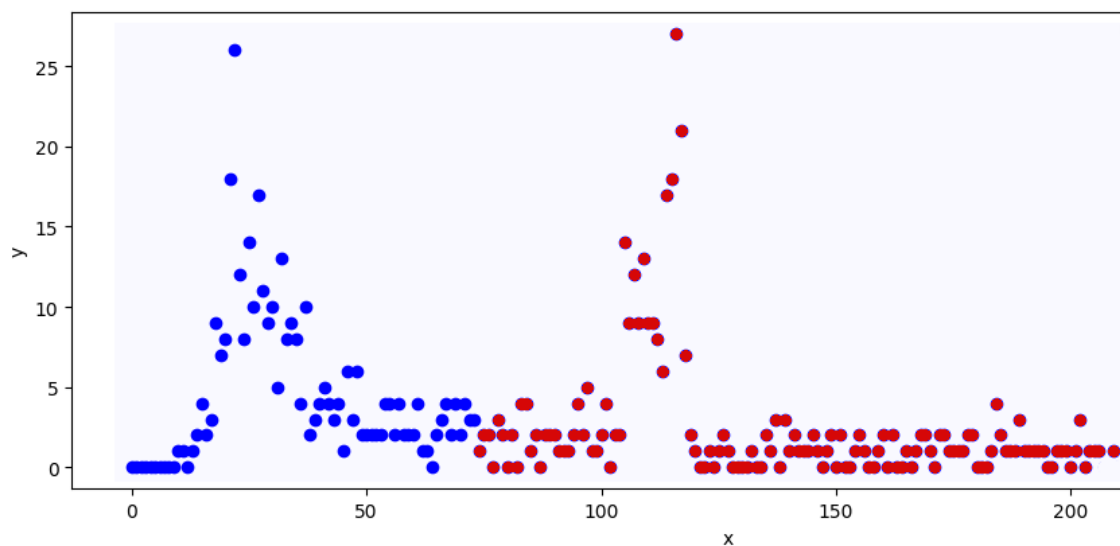


Рисунок 3.3 – Центри знайдені алгоритмом MoG

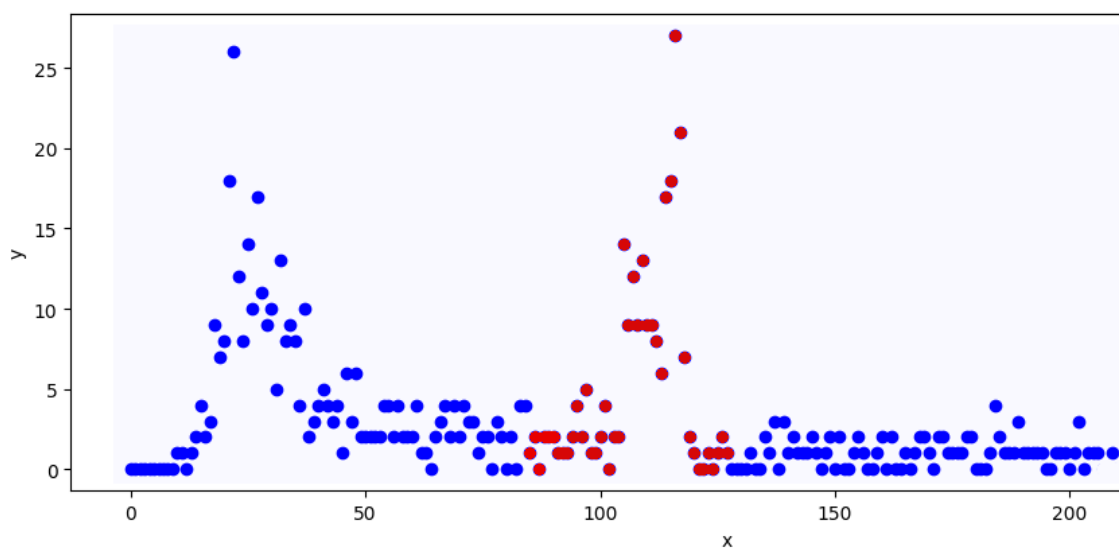


Рисунок 3.4 – Уточнений розподіл запропонованим алгоритмом

Так результат роботи виглядає як уточнення роботи MoG з вказівкою який з розподілів є фоном.

### 3.3 Аналіз результатів



Рисунок 3.5 – Розбиття на класи

Таким чином, було вирішено декілька головних проблем, а саме, як показано на рисунку 3.4, запропонована модель виділила фоном (синій) клас з меншою дисперсією, а не той, що має більшу апріорну ймовірність, тобто зустрічається більшу частину часу. Так, алгоритм є більш стійким до повільно рухаючихся об'єктів.

Також з рисунку 3.4 видно, що виправилася ситуація з частковим відтягуванням центрів розподілів.

До того ж, повністю виправилась ситуація, представлена на рисунку 2.10. Порівняння роботи алгоритмів MOG, MOG2 та запропонованого.

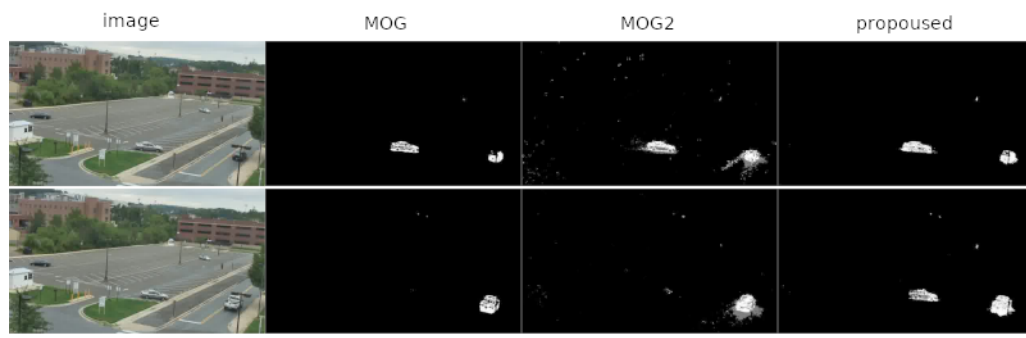


Рисунок 3.6 – Порівняння роботи алгоритмів

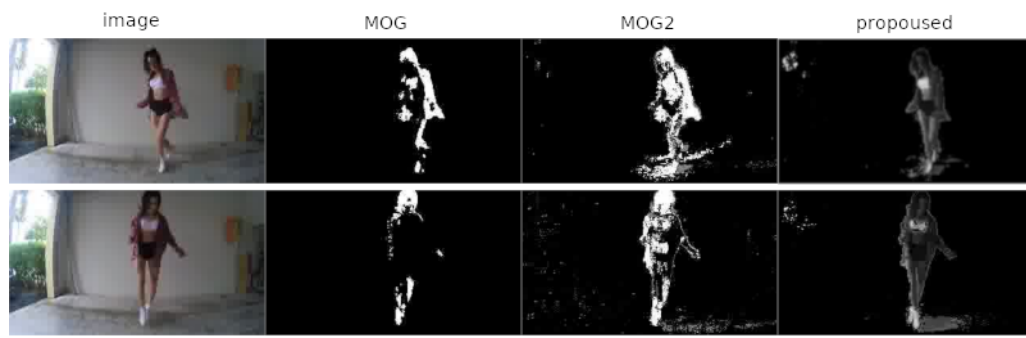


Рисунок 3.7 – Порівняння роботи алгоритмів

### Висновки до розділу 3

У цьому розділі був запропонований алгоритм для відновлення параметрів фону, який базується на проаналізованій у попередньому розділі моделі MoG. Запропонований підхід дозволяє значно покращити процес відновлення фону для багатьох ситуацій на відеозображенні.

## ВИСНОВКИ

В результаті виконання роботи було:

- 1) Проаналізовано відомі методи розділення та кластеризації зображення, а також відновлення статистичних параметрів фону.
- 2) Показано, що більшість алгоритмів детекції руху не застосовні до задачі відновлення фону, а методи кластеризації потребують модифікацій для їх застосування.
- 3) Реалізовано відому модель MoG з застосуванням ЕМ та знайдено 3 критичні ситуації, при яких якість відновлення параметрів фону не задовільна.
- 4) Запропоновано нову модель на основі MoG, що включає в себе клас рівномірного розподілу.
- 5) Досліджено запропоновану модель та проведено порівняння з реалізаціями MoG в відкритій бібліотеці OpenCV.

Так дослідження показало, що запропонована модель повністю вирішує дві з трьох критичних для MoG ситуації без додаткового використання попередньої та після обробки вхідного відеозображення. А саме ситуації з єдиним спостереженням об'єкту та без нього взагалі.

Частково вирішена проблема відносно статичного об'єкту, так вимога до відносно високої швидкості переміщення по кадру об'єкта стала значно меншою, проте якщо він зупиняється через половину часу від початку спостереження об'єкт гарантовано почне сприйматися як фон.

Недоліками є дуже висока ресурсомісткість, та висока чутливість до початкових параметрів розподілу.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Stauffer C. Adaptive background mixture models for real-time tracking [Текст]/ C. Stauffer, W. Grimson // Proc IEEE Conf on Comp Vision and Patt Recog. - 1999. - С. 246-252.
2. Lee B. Background estimation for video surveillance. Image and vision computing [Текст]/ B. Lee, M. Hedley //New Zealand. -2002. -С. 315-320.
3. Zheng J. Extracting roadway background image: A mode based approach. [Текст]/ J. Zheng, Y. Wang, N. Nihan , E. Hallenbeck // J Transport Res Report. -2006. -С. 82-88.
4. Mika A. Foreground-Background Segmentation Based on Codebook and EdgeDetector. [Текст]/ A. Mika, E. Mousse // -2014.
5. McFarlane N. Segmentation and tracking of piglets in images. [Текст]/ N. McFarlane // British Machine Vision and Applications. -1995. -С. 187–193.
6. Culibrk H. Neural Network Approach to Background Modeling for Video Object Segmentation. [Текст]/ D. Socek, H. Kalva, B. Furht.// IEEE Transactions on Neural Networks. -2007.
7. Elgammal A. Non-parametric model for background subtraction. [Текст]/ D. Harwood, L. Davis, A. Elgammal. // 6th European Conf on Computer Vision (ECCV 2000), Dublin, Ireland. -2000. -С. 751-767.
8. Zhang H. Fusing Color and Texture Features for Background Model. [Текст]/H. Zhang // Third Int Conf on Fuzzy Systems and Knowledge Discovery. -2006. -С. 887-893
9. Sigari M. Fuzzy running average and fuzzy background subtraction: concepts and application. [Текст]/ M. Sagiri, N. Mozayani. // Int J Comput Sci Network Security. -2008; -С. 138-143.
10. Krumm J. Wallflower: Principles and practice of background maintenance. [Текст]/ J. Krumm, B. Brumitt, B. Mayers. // Int Conf on Computer Vision, (ICCV 1999), Corfu, Greece. -1999; -С. 255-261.
11. Chang R. Vision modules for a multi sensory bridge monitoring approach.

[Текст]/ M. Trivedi, T. Ghandi, R. Chang. // ITSC 2004, October 2004; -C. 971-976.

12. Raquel U. Mixtures of Gaussians and EM. [Текст]/ U. Raquel, R. Zemel. // CSC 411. -2015.

13. Birgi T. Background Subtraction. [Текст]/ T. Birdi. // -2009.

14. Dhanachandra N. Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. [Текст]/ K. Manglem, Y. Chau, N. Dhanachandra. // Procedia Computer Science. -2015. -C. 764–771.

15. K. Kim. Real-time foreground-background segmentation using codebookmodel. [Текст]/ Kim K., Harwood D., Davis L. // Real-Time Imaging. -2005. -C. 167-256.

16. W. X. Kang. The Comparative Research on Image Segmentation Algorithms. [Текст]/ Kang W. X., Yang Q. Q., Liang R. R.// IEEE Conference on ETCS. -2009. -C. 703-707.

17. N. Singla. Motion Detection Based on Frame DifferenceMethod [Текст]/ Singla N. // International Journal of Information Computation Technology Volume 4, Number 15. -2014. -C. 1559-1565

18. R. Liu. Background Subtraction with Multispectral Images Using Codebook Algorithm. [Текст]/ Liu, R., Ruichek Y. // Lecture Notes in Computer Science. -C. 581–590.

19. J. Zhou. Color Separation for Background Subtraction. [Текст]/ J. Zhou. // -2016.